

**Technical Report OSU-CISRC-5/15-TR06**

Department of Computer Science and Engineering

The Ohio State University

Columbus, OH 43210-1277

Ftpsite: **ftp.cse.ohio-state.edu**

Login: **anonymous**

Directory: **pub/tech-report/2015**

File: **TR06.pdf**

Website: **http://www.cse.ohio-state.edu/research/techReport.shtml**

## **Boosting Contextual Information for Deep Neural Network Based Voice Activity Detection**

**Xiao-Lei Zhang**

Department of Computer Science and Engineering  
The Ohio State University, Columbus, OH 43210, USA  
*xiaolei.zhang9@gmail.com*

**DeLiang Wang**

Department of Computer Science and Engineering & Center for Cognitive and Brain Sciences  
The Ohio State University, Columbus, OH 43210, USA  
*dwang@cse.ohio-state.edu*

*Abstract* – Voice activity detection (VAD) is an important topic in audio signal processing. Contextual information is important for improving the performance of VAD at low signal-to-noise ratios. Here we explore contextual information by machine learning methods at three levels. At the top level, we employ an ensemble learning framework, named multi-resolution stacking (MRS), which is a stack of ensemble classifiers. Each classifier in a building block inputs the concatenation of the predictions of its lower building blocks and the expansion of the raw acoustic feature by a given window (called a resolution). At the middle level, we describe a base classifier in MRS, named boosted deep neural network (bDNN). bDNN first generates multiple base predictions from different contexts of a single frame by only one DNN and then aggregates the base predictions for a better prediction of the frame, and it is different from computationally-expensive boosting methods that train ensembles of classifiers for multiple base predictions. At the bottom level, we employ the multi-resolution cochleagram feature, which incorporates the contextual information by concatenating the cochleagram features at multiple spectrotemporal resolutions. Experimental results show that the MRS-based VAD outperforms other VADs by a considerable margin. Moreover, when trained on a large amount of noise types and a wide range of signal-to-noise ratios, the MRS-based

VAD demonstrates surprisingly good generalization performance on unseen test scenarios, approaching the performance with noise-dependent training.

*Index Terms* – Boosting, cochleagram, deep neural network, multi-resolution stacking, voice activity detection.

# Boosting Contextual Information for Deep Neural Network Based Voice Activity Detection

Xiao-Lei Zhang, *Member, IEEE* and DeLiang Wang, *Fellow, IEEE*

**Abstract**—Voice activity detection (VAD) is an important topic in audio signal processing. Contextual information is important for improving the performance of VAD at low signal-to-noise ratios. Here we explore contextual information by machine learning methods at three levels. At the top level, we employ an ensemble learning framework, named multi-resolution stacking (MRS), which is a stack of ensemble classifiers. Each classifier in a building block inputs the concatenation of the predictions of its lower building blocks and the expansion of the raw acoustic feature by a given window (called a resolution). At the middle level, we describe a base classifier in MRS, named boosted deep neural network (bDNN). bDNN first generates multiple base predictions from different contexts of a single frame by only one DNN and then aggregates the base predictions for a better prediction of the frame, and it is different from computationally-expensive boosting methods that train ensembles of classifiers for multiple base predictions. At the bottom level, we employ the multi-resolution cochleagram feature, which incorporates the contextual information by concatenating the cochleagram features at multiple spectrotemporal resolutions. Experimental results show that the MRS-based VAD outperforms other VADs by a considerable margin. Moreover, when trained on a large amount of noise types and a wide range of signal-to-noise ratios, the MRS-based VAD demonstrates surprisingly good generalization performance on unseen test scenarios, approaching the performance with noise-dependent training.

**Index Terms**—Boosting, cochleagram, deep neural network, multi-resolution stacking, voice activity detection.

## I. INTRODUCTION

VOICE activity detection (VAD) is an important preprocessor for many audio signal processing systems. For example, it improves the efficiency of speech communication systems [2] by detecting and transmitting only speech signals. It helps speech enhancement algorithms [3] and speech recognition systems [9], [13] by filtering out silence and noise segments. One of the major challenging problems of VAD is to make it perform in low signal-to-noise ratio (SNR) environments. Early research focused on signal processing based acoustic features, including energy in the time domain, pitch detection, zero-crossing rate, and several spectral energy based features such as energy-entropy, spectral correlation, spectral divergence, higher-order statistics [25]. Recent development includes low-frequency ultrasound [22] and single frequency filtering [1]. Exploring feature is important in improving VAD research from the aspect of acoustic mechanism. However, each acoustic feature reflects only some characteristics of human voice. Moreover, using the features independently is not

very effective in extremely difficult scenarios. Hence, fusing the features together as the input of some data-driven methods may be an effective usage of the features for improving the overall performance of VAD.

Another important research branch of VAD is statistical signal processing. These techniques make model assumptions on the distributions of speech and background noise (usually in the spectral domain) respectively, and then design statistical algorithms to dynamically estimate the model parameters. Typical model assumptions include the Gaussian distribution [34], [43], Laplace distribution [16], Gamma distribution [6], or their combinations [6]. The most popular parameter estimation method is the minimum mean square error estimation [12]. In addition, long-term contextual information is shown to be useful in improving the performance [30]. Due to the simplicity of the model assumptions and online updating of the parameters, this kind of methods may generate reasonable results in various noise scenarios. In many cases, they work better than energy based methods. But statistical model based methods have limitations. First, model assumptions may not fully capture global data distributions, since the models usually have too few parameters and they estimate parameters on-the-fly from limited local observations. Second, with relatively few parameters, they may not be flexible enough in fusing multiple acoustic features. Moreover, most methods update parameters during the pure noise phase which may cause them fail when the noise changes rapidly during the voice phase.

The third popular branch of VAD research is machine learning methods, which train acoustic models from given noisy corpora and apply the models to real-world test environments. They have two main research objectives: one is to improve the *discriminative ability* of models when the noise scenarios of training and test corpora are matching; the other is to improve the *generalization ability* (i.e. detection accuracy) of models to test noise scenarios when the test noise scenarios are unseen from or mismatching with the training noise scenarios.

Most machine learning methods focus on how to improve the discriminative ability. We summarize them briefly as follows. In terms of whether their training corpora are manually labeled, they can be categorized to *unsupervised learning* which uses unlabeled training corpora, or *supervised learning* which uses labeled training corpora. Many unsupervised methods belong to dimensionality reduction, which first extract noise-robust low-dimensional features from highly-variant high-dimensional observations and then apply the features to classifiers. They include principle component analysis [31], non-negative matrix factorization [36], and spectral decomposition of graph Laplacian [23]. Some methods

Xiao-Lei Zhang and DeLiang Wang are with the Department of Computer Science & Engineering and Center for Cognitive & Brain Sciences, The Ohio State University, Columbus, OH, USA (e-mail: xiaolei.zhang9@gmail.com, dwang@cse.ohio-state.edu).

use clustering algorithms directly, such as  $k$ -means clustering [17] and Gaussian mixture models [31]. Unsupervised methods are able to explore multiple features and train robust models from vast amount of recorded data, however, when the tasks are too difficult that most speech signal is drowned in background noise, such as babble noise with an SNR below 0 dB, unsupervised methods are helpless. Note that, statistical signal processing based VADs can also be regarded as unsupervised methods, which train models from a few local observations and accumulated historical information.

Supervised learning methods take VAD as a binary-class classification problem—speech or non-speech. The techniques can be roughly categorized to four classes: probabilistic models, kernel methods, neural networks, and ensemble methods. Probabilistic models include Gaussian mixture models [26] and conditional random fields [36]. Kernel methods mainly include various support vector machines (SVM), such as [11], [33]. These two kinds cannot handle large-scale corpora well, so that they are difficult to be used in practice since we need large-scale training corpora to cover rather complicated real-world noisy environments.

Recently, deep neural networks (DNN) and their extensions [14], [21], [32], [38], [45], [47], which have a strong scalability to large-scale corpora, showed good performance in extremely difficult scenarios and are competitive in real-world applications. Specifically, in [45], Zhang and Wu proposed to apply standard deep belief networks to VAD and reported better performance than SVM, where the networks were pretrained as in [9]. In [47], Zhang and Wang further proposed to generate multiple different predictions from a single DNN by boosting contextual information and reported significant improvement over the standard DNN in difficult noise scenarios and low SNR levels. In [14], [21], the authors applied deep recurrent neural networks to capture historical contextual information and reported significant improvement over Gaussian mixture models and statistical signal processing methods. However, the performance improvements of the aforementioned DNN methods were observed when the DNNs were trained *noise-dependently*, i.e. the noise scenarios of training and test are matching. When applying DNN-based VADs to unseen test scenarios, the performance dropped significantly as shown in [14], [46]. Recently, in [32], [38], the authors trained DNN and convolutional neural networks together from large-scale real-world data [39] and demonstrated impressive two-phase improvements. However, because each model in [32], [38] were binded to a given channel, we still do not know exactly how the models will generalize to different noise scenarios. Due to the restriction of the task setting, the results do not have a quantitative evaluation on how the models vary with SNR levels, which need a further investigation.

To summarize, DNN-based VADs with noise dependent training have demonstrated good performance and have shown strong potential in practice. In this paper, we further develop DNN-based VADs by exploring contextual information heavily in three novel levels. Motivated by recent progress of speech separation [41], [42], we also investigate quantitatively how DNN-based VADs can generalize to unseen test noise scenarios with the variation of SNR through noise-independent

training. The main contributions of this paper are summarized as follows:

- **Multi-resolution stacking (MRS).** MRS is a stack of ensemble classifiers. Each classifier in a building block takes the concatenation of the soft output predictions of the lower building block and the expansion of the original acoustic feature in a window (called a resolution). The classifiers in the same building block have different resolutions, which is the novelty of this framework.
- **Boosted deep neural network (bDNN).** bDNN is proposed as the base classifier of MRS. It first generates multiple base predictions on a frame by boosting the contextual information of the frame, and then aggregates the base predictions for a stronger one. bDNN generates multiple predictions from a single DNN, which is its novelty compared to ensemble DNNs. Preliminary results [47] showed that it can significantly outperform DNN-based VAD without increasing computational complexity.
- **Multi-resolution cochleagram (MRCG) feature.** MRCG [7], which was first proposed for speech separation, is employed as a new acoustic feature for VAD. It concatenates multiple cochleagram features calculated at different spectral and temporal resolutions.
- **Noise-independent training.** We train the proposed method with a corpus that has a vast amount of noise scenarios with a wide variation of SNR levels, and test it in unseen and difficult noise scenarios. We find that the method can generalize well.

Empirical results on the AURORA2 [28] and AURORA4 corpora [29] show that the MRS-based VAD outperforms a DNN-based VAD [45] and 4 other comparison methods. Moreover, when the proposed method is trained noise-independently, its performance on unseen test noise scenarios at various SNR levels is surprisingly as good as the proposed method with noise-dependent training. This paper differs from our preliminary work [47] in several major aspects, which include the use of MRS and noise-independent training in this paper (but not in [47]) and new parameter settings for bDNN and MRCG. Consequently, experimental results in this paper are different from those reported in [47].

The paper is organized as follows. In Section II, we introduce the MRS framework. In Section III, we present the bDNN model. In Section IV, we introduce the MRCG feature. In Section V, we present results with noise-dependent training. In Section VI, we present results with noise-independent training. Finally, we conclude in Section VII.

## II. MULTI-RESOLUTION STACKING

We formulate VAD as a supervised classification problem. Specifically, a long speech signal is divided to multiple short-term overlapped frames, each of which ranges usually from 10 to 25 milliseconds. For a classification problem, each frame  $\mathbf{o}_m$  in the time domain is transformed to an acoustic feature in the spectral domain, denoted as  $\mathbf{x}_m$ , where  $m = 1, \dots, M$  indexes the time of the frame. To construct a training set, the frame  $\mathbf{x}_m$  is manually labeled as  $y_m = 1$  or  $y_m = 0$ , indicating  $\mathbf{x}_m$  is a speech or noise frame respectively. A classifier  $f(\cdot)$

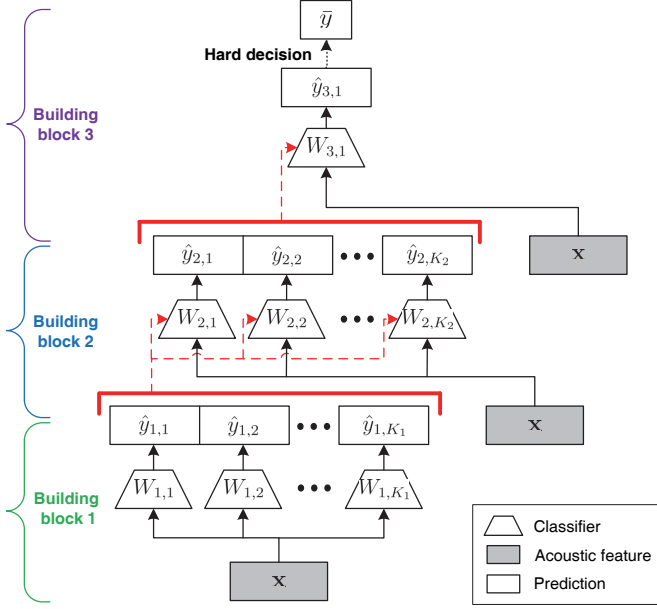


Fig. 1. Principle of multi-resolution stacking. The soft predictions of all base classifiers in a building block are combined in the red line as part of the input of the base classifiers in the upper building block. The input of a base classifier is the concatenation of the soft predictions from the lower building block and the acoustic feature that is extended by a window.

is trained on  $\{(\mathbf{x}_m, y_m)\}_{m=1}^M$  and tested on a different set  $\{\mathbf{x}_n\}_{n=1}^N$ .

It is known that contextual information is important in improving the performance of VAD. One common technique to incorporate contextual information is to train models with a fixed window length that performs the best among several choices of window lengths. We denote the technique of adding a window to incorporate neighboring frames the *resolution*. Here, we argue that (i) for a certain task, although only one resolution performs the best, other resolutions may still provide useful information that may further improve the performance; (ii) although we can manage to pick the best resolution for a certain task, it is still inconvenient to do so case by case. We propose a simple framework, named *multi-resolution stacking*, to solve the two problems together.

#### A. Training Phase of MRS

As described in Fig. 1, MRS is a stack of classifier ensembles. In the training stage of MRS, suppose we are to train  $S$  building blocks ( $S = 3$  in Fig. 1). The  $s$ th building block has  $K_s$  classifiers, denoted as  $\{f_{s,k}(\cdot)\}_{k=1}^{K_s}$ , each of which has a predefined resolution  $W_{s,k}$ . The  $k$ th classifier  $f_{s,k}(\cdot)$  takes vector  $\mathbf{z}_{s,k}$  as the input:<sup>1</sup>

$$\mathbf{z}_{s,k} = \begin{cases} \mathbf{x}'_{s,k} & \text{if } s = 1 \\ [\hat{y}_{s-1,1}, \dots, \hat{y}_{s-1,K_{s-1}}, \mathbf{x}'_{s,k}]^T & \text{if } s > 1 \end{cases} \quad (1)$$

and takes  $y$  as the training target, where  $\{\hat{y}_{s-1,k'}\}_{k'=1}^{K_{s-1}}$  are the soft predictions of  $\mathbf{x}$  produced by the  $(s-1)$ th building block

and  $\mathbf{x}'_{s,k}$  is an expansion of  $\mathbf{x}$  given the resolution  $W_{s,k}$ :

$$\mathbf{x}'_{s,k} = [\mathbf{x}_{-W_{s,k}}^T, \mathbf{x}_{-W_{s,k}+1}^T, \dots, \mathbf{x}_0^T, \dots, \mathbf{x}_{W_{s,k}-1}^T, \mathbf{x}_{W_{s,k}}^T]^T \quad (2)$$

where the subscript 0 in  $\mathbf{x}_0$  is a general index for describing any training frame. After  $f_{s,k}(\cdot)$  is trained, it produces a soft prediction  $\hat{y}_{s,k}$  of  $\mathbf{x}$  for the upper building block.

The resolution  $W$  will double the size of training data, therefore, MRS is hard to handle both a large  $W$  and a large training set. To reduce the memory requirement of computing power, we present a trick: one can pick a subset of frames within the window instead of all frames. In this paper, we expand  $\{-W, -W+u, -W+2u, \dots, -1-u, -1, 0, 1, 1+u, \dots, W-2u, W-u, W\}$ ,

$$\mathbf{x}'_{s,k} = [\mathbf{x}_{-W_{s,k}}^T, \mathbf{x}_{-W_{s,k}+u}^T, \dots, \mathbf{x}_0^T, \dots, \mathbf{x}_{W_{s,k}-1}^T, \mathbf{x}_{W_{s,k}}^T]^T \quad (3)$$

where  $u$  is a user defined integer parameter. This trick not only makes all classifiers in a building block have the same amount of memory requirement but also does not decrease the performance significantly in experience.

#### B. Test Phase of MRS

In the test stage of MRS, we get a serial soft predictions as we did in the training stage from the bottom stack to the top stack. After getting the output of the  $S$ th building block which contains only one classifier as shown in Fig. 1, we do a hard decision on the output, e.g.  $\hat{y}_{S,1}$ , by:

$$\bar{y} = \begin{cases} 1 & \text{if } \hat{y}_{S,1} \geq \delta \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

and take  $\bar{y}$  as the final prediction of  $\mathbf{x}$ , where  $\delta$  is a decision threshold tuned on a development set.

#### C. On the Related Work and Novelty of MRS

In noise-robust speech signal processing, using the optimal resolution is a common technique, such as statistical signal processing based VADs [30] and recent machine learning based VADs. However, the models with suboptimal resolutions may also provide useful information. Fusing ensemble models is another common technique, such as fusing DNN and convolutional neural networks in [32], [38], but, they do not consider different resolutions and stacking, and do not take the raw feature as the input of the consensus model. Stacking ensemble classifiers has been used in speech separation [27] and recognition [44], but they did not consider different resolutions. To summarize, stacking ensembles of classifiers in different resolutions are the novelty of the framework.

<sup>1</sup>For clarity, we omit the time index of the training and test frames from here to the end of this section.

### III. BOOSTED DNN

In this section, we fill MRS by a strong base classifier—boosted DNN. We first present the bDNN algorithm in Sections III-A, then introduce the motivation of bDNN in Section III-C, and our DNN model in Section III-B. Finally, we present the novelty of the bDNN model in Section III-D.

Deep neural network is a strong classifier that can approach to the minimum expectation risk—the ideal minimum risk given the infinite amount of training data—when the input data is large scale. It has been adopted in recent VAD studies. One common technique to further improve the prediction accuracy of DNN is ensemble learning, which trains multiple DNNs that yield different base predictions, such that when the base predictions are aggregated, the final prediction is boosted to be better than any of the base predictions. However, it is too expensive to train a set of DNNs if they do not receive significantly different knowledge from the input. To alleviate the computational load but benefit from ensemble learning, we proposed bDNN, which can generate multiple different base predictions on a single frame by training only one DNN.

#### A. Boosted DNN

In the training stage of bDNN, we first expand each speech frame by Eq. (2). The main difference between training bDNN and DNN is that for bDNN, we further expand  $y_m$  to

$$\mathbf{y}'_m = [y_{m-W}, y_{m-W+1}, \dots, y_m, \dots, y_{m+W-1}, y_{m+W}]^T \quad (5)$$

That is to say, bDNN is a DNN model that is trained on a new corpus  $\{(\mathbf{x}'_m, \mathbf{y}'_m)\}_{m=1}^M$ , and has  $(2W+1)d$  input units and  $2W+1$  output units, where  $d$  is the dimension of  $\mathbf{x}_m$ .

In the test stage of bDNN, we aim to predict the label of frame  $\mathbf{x}_n$ , which consists of three steps as shown in Fig. 2. The first step expands  $\mathbf{x}_n$  to a large observation  $\mathbf{x}'_n$  as done in the training phase, so as to get a new test corpus  $\{\mathbf{x}'_n\}_{n=1}^N$  (Fig. 2A). The second step gets the  $(2W+1)$ -dimensional prediction of  $\mathbf{x}'_n$  from the DNN, denoted as  $\mathbf{y}'_n = [y_{n-W}^{(-W)}, y_{n-W+1}^{(-W+1)}, \dots, y_n^{(0)}, \dots, y_{n+W-1}^{(W-1)}, y_{n+W}^{(W)}]^T$  (Fig. 2B). The third step aggregates the results to predict the soft decision of  $\mathbf{x}_n$ , denoted as  $\hat{y}_n$  (Fig. 2C):

$$\hat{y}_n = \frac{\sum_{w=-W}^W y_n^{(w)}}{2W+1} \quad (6)$$

Finally, we make a hard decision by

$$\bar{y}_n = \begin{cases} 1 & \text{if } \hat{y}_n \geq \eta \\ -1 & \text{otherwise} \end{cases} \quad (7)$$

where  $\eta \in [-1, 1]$  is a decision threshold tuned on a development set according to some predefined performance measurement.

Note that (i) when we adopt the trick in Section II to alleviate the memory requirement, Eq. (5) should be modified accordingly as follows:

$$\mathbf{y}'_m = [y_{m-W}, y_{m-W+u}, \dots, y_m, \dots, y_{m+W-u}, y_{m+W}]^T \quad (8)$$

(ii) When bDNN is used in MRS, the input of bDNN should be adapted to Eq. (1) which has  $(2W+1)d + K_s$  dimensions.

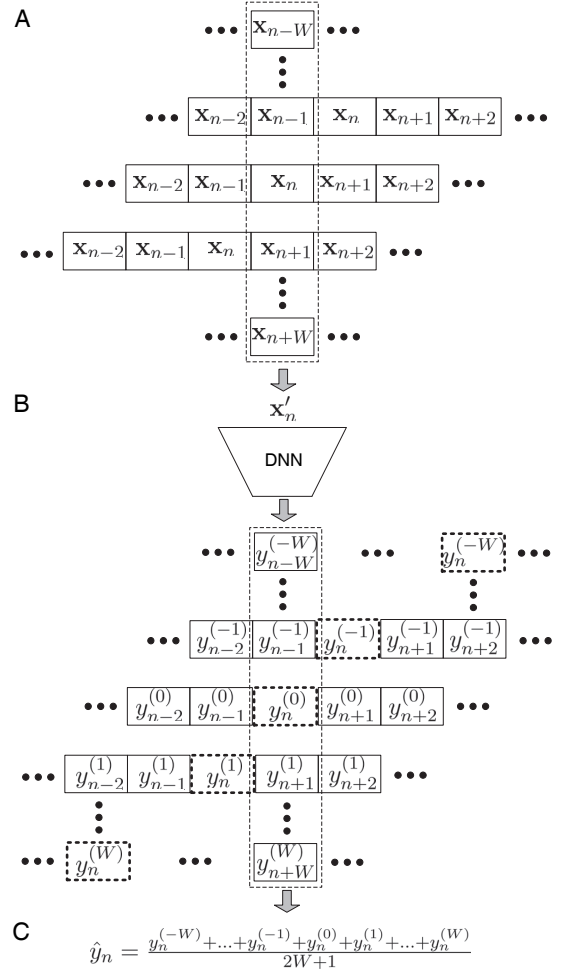


Fig. 2. Test phase of bDNN. (A) Expanding  $\mathbf{x}_n$  to a new feature (included in the dashed rectangle, denoted as  $\mathbf{x}'_n$ ) given the half-window size  $W$ . (B) Predicting labels of  $\mathbf{x}'_n$  to yield a  $(2W+1)$ -dimensional vector (included in the dashed rectangle) by DNN. (C) Aggregating the prediction results by the given equation from the soft output units drawn in the bold dashed rectangles of Fig. 1b.

#### B. DNN Model

We adopt contemporary DNN training methods, and use the *area under the receiver operating characteristic curve* (AUC) as the performance metric for selecting the best DNN model in the training process.

The template of deep models is described as follows:

$$\mathbf{y} = f \left( h_{(L)} \left( \dots h_{(l)} \left( \dots h_{(2)} \left( h_{(1)} \left( \mathbf{x}^{(0)} \right) \right) \right) \right) \right) \quad (9)$$

where  $l$  denotes the  $l$ th hidden layer from the bottom,  $h_{(l)}(\cdot)$  is the nonlinear mapping function of the  $l$ th layer, and  $\mathbf{x}^{(0)}$  is the input feature vector. Different from [45], we use the rectified linear unit  $y = \max(0, x)$  for hidden layers, sigmoid function  $y = \frac{1}{1+e^{-x}}$  for the output layer, and a dropout strategy to specify the DNN model [8]. See Section III-D2 for a detailed explanation. In addition, we employ the adaptive stochastic gradient descent [10] and a momentum term [35] to train the DNN. These training schemes accelerate traditional gradient descent training and facilitate large-scale parallel computing. Note that no pretraining is used in our DNN training.

---

**Algorithm 1** AUC calculation.

---

**Input:** Number of training data points  $n$ , manual label vector  $\mathbf{y} = [y_1, \dots, y_n]^T$ , and predicted soft values  $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_n]^T$

**Initialization:**  $a = 0, b = 0, \text{swapped\_pairs} = 0$

**Output:** AUC  $A$

```

1: Sort  $\hat{\mathbf{y}}$  in descending order, denoted as  $\hat{\mathbf{y}}^*$ , and reorder  $\mathbf{y}$ 
   along with  $\hat{\mathbf{y}}$ , denoted as  $\mathbf{y}^*$ 
2: for  $i = 1, \dots, n$  do
3:   if  $y_i^* > 0$  then
4:      $\text{swapped\_pairs} \leftarrow \text{swapped\_pairs} + b$ 
5:      $a \leftarrow a + 1$ 
6:   else
7:      $b \leftarrow b + 1$ 
8:   end if
9: end for
10:  $A = 1 - \frac{\text{swapped\_pairs}}{ab}$ 

```

---

AUC can be calculated efficiently by Algorithm 1. The reasons why we use AUC as the performance metric are as follows. First, AUC measures the receiver operating characteristic (ROC) curve quantitatively. The ROC curve is considered as an overall metric of the VAD performance rather than the detection accuracy, since the speech-to-nonspeech ratio is usually imbalanced, and also since one usually tunes the decision threshold of VAD for specific applications. Second, AUC matches the metric of *speech hit rate minus false alarm rate* (HIT-FA) which is defined as:

$$\begin{aligned} \text{HIT} - \text{FA} &= \text{Hit rate} - \text{false alarm rate} \\ &= \frac{\# \text{correctly predicted speech frames}}{\# \text{ground-truth speech frames}} \\ &\quad - \frac{\# \text{wrongly predicted noise frames}}{\# \text{ground-truth noise frames}} \end{aligned}$$

HIT-FA reflects the performance of VAD at the optimal operating point of its tunable decision threshold.

### C. Motivation of Boosted DNN

Originally, we planed to first train multiple DNNs  $g_{-W}(\cdot), \dots, g_W(\cdot)$  and then aggregate the outputs of the DNNs. Each DNN trains a mapping function from an expansion of the input  $\mathbf{x}_m$  to its manual label  $y_m$  in the standard way, but the expansion in different DNNs is a sliding window around  $\mathbf{x}_m$ . For example, the  $i$ th DNN  $g_i(\cdot)$  takes  $\mathbf{x}_m^{(i)} = [\mathbf{x}_{m-W+i}^T, \dots, \mathbf{x}_m^T, \dots, \mathbf{x}_{m+i}^T, \dots, \mathbf{x}_{m+W+i}^T]^T$  as its input and outputs the prediction  $y_m^{(i)}$ . The ensemble method gets the final prediction  $\hat{y}_m$  by aggregating the base predictions  $\hat{y}_m = \sum_{i=-W}^W y_m^{(i)} / (2W + 1)$ .

After observing the fact that  $[\mathbf{x}_{m-W}^T, \dots, \mathbf{x}_m^T, \dots, \mathbf{x}_{m+W}^T]^T$  appears as the expanded feature of  $\mathbf{x}_{m-i}$  for training  $y_m^{(i)} = g_i(\cdot)$  where  $i = -W, \dots, W$ , we propose to integrate the outputs  $y_m^{(i)}$  together and train a new DNN model:

$$\begin{bmatrix} y_{m-W}^{(-W)} \\ \vdots \\ y_{m+W}^{(W)} \end{bmatrix} = g \left( \begin{bmatrix} \mathbf{x}_{m-W} \\ \vdots \\ \mathbf{x}_{m+W} \end{bmatrix} \right) \quad (10)$$

where  $g(\cdot)$  is the DNN model of bDNN that has multiple output units. Then, we propose to aggregate the base predictions for the final prediction as in Eq. (6). The main difference between bDNN and the aforementioned inefficient method is that the base predictions  $y_m^{(i)}$  of bDNN share the same parameters of the hidden units of a single DNN model, while the base predictions of the inefficient method are generated independently from multiple DNN models. bDNN saves the computational load greatly with some loss of flexibility of model training.

### D. On the Related Work and Novelty of the Boosted DNN

1) *On the Relationship Between Boosted DNN and the Common Boosting Techniques:* For bDNN, the output target of the  $m$ th frame, i.e.  $y_m$ , is assumed to be generated from  $[\mathbf{x}_{m-2W}^T, \dots, \mathbf{x}_m^T, \dots, \mathbf{x}_{m+2W}^T]^T$ . bDNN generates multiple base predictions of  $y_m$  by extracting part of the input feature. The method of manipulating the input feature only is related to but different from bagging [4] and Adaboost [15] which manipulate the input data set, and is also different from random forests [5] which manipulates the input data set and features together. We also tried to generate base predictions from different subsets of data, but we found that the performance was not as good as the performance produced from the entire data set due to the performance decrease of each base classifier.

2) *On the Difference Between the bDNN Based VAD and the DNN Based VAD in [45]:* The output targets of bDNN and the method in [45] are different. bDNN reformulates VAD as a structural learning problem that learns an encoder that maps a serial concatenation of multiple frames to a binary code, while the method in [45] takes VAD as a traditional binary-class classification problem that predicts the classes of frames in sequence. The structural learning fully utilizes the contextual information of the output target.

The DNN implementations of bDNN and the method in [45] are also different in respect of the network structure and training method. (i) bDNN adopts dropout [19] as the regularization to prevent DNN from overfitting, while the DNN model in [45] does not use any regularization method. Dropout randomly inactivates the (hidden and/or input) units. Due to such a regularization, bDNN is able to train much larger model with a stronger generalization ability than the DNN model in [45]. (ii) bDNN uses the rectified linear unit  $y = \max(0, x)$  as the hidden unit and sigmoid function  $y = \frac{1}{1+e^{-x}}$  as the output unit, while the method in [45] uses the sigmoid function as the hidden unit and softmax function  $y_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$  as the output unit. Rectified linear unit can be trained faster than the sigmoid function and helps DNN learn local patterns better. (iii) bDNN does not use pretraining [18] while the method in [45] uses pretraining, where pretraining is known to prevent bad local minima of DNN. Recent results show that when the data set is large enough, the performance of DNN without pretraining is also good enough.

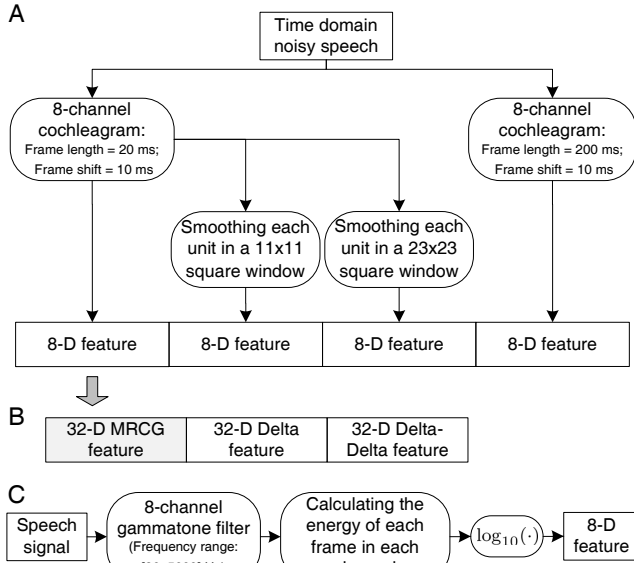


Fig. 3. The MRCG feature. (A) Diagram of the process of extracting a 32-dimensional MRCG feature. “ $(2W + 1) \times (2W + 1)$  square window” means that the value of a given time-frequency unit is replaced by the average value of its neighboring units that fall into the window centered at the given unit and extending in the axes of time and frequency. (B) Expanding MRCG to a 96-dimensional feature that consists of the original MRCG feature, its Delta feature and Delta-Delta feature. (C) Calculation of the 8-dimensional cochleagram features in detail.

#### IV. MRCG FEATURE

In this section, we introduce the MRCG feature which was first proposed in [7] for speech separation.<sup>2</sup> The key idea of MRCG is to incorporate both local information and global information through multi-resolution extraction. The local information is produced by extracting cochleagram features with a small frame length and a small smoothing window (i.e., high resolutions). The global information is produced by extracting cochleagram features with a large frame length or a large smoothing window (i.e., low resolutions). It has been shown that cochleagram features with a low resolution, such as frame length = 200 ms, can detect patterns of noisy speech better than that with only a high resolution, and features with high resolutions complement those with low resolutions. Therefore, concatenating them together is better than using them separately.

As illustrated in Fig. 3A, MRCG is a concatenation of 4 cochleagram features with different window sizes and different frame lengths. The first and fourth cochleagram features are generated from two  $U$ -channel gammatone filterbanks ( $U = 8$  in this paper) with frame lengths set to 20 ms and 200 ms respectively. The second and third cochleagram features are calculated by smoothing each time-frequency unit of the first cochleagram feature with two square windows that are centered on the unit and have the sizes of  $11 \times 11$  and  $23 \times 23$ . Because the windows on the first and last few channels (or frames) of the two cochleagram features may overflow, we cut off the overflowed parts of the windows. Note that the multi-resolution strategy is a common technique not limited

to the cochleagram feature [20], [24].

After calculating the  $(4 \times U)$ -dimensional MRCG feature, we further calculate its Deltas and double Deltas, and then combine all three into a  $(12 \times U)$ -dimensional feature (Fig. 3B). A Delta feature is calculated by

$$\Delta x_n = \frac{(x_{n+1} - x_{n-1}) + 2(x_{n+2} - x_{n-2})}{10} \quad (11)$$

where  $x_k$  is the  $k$ th unit of MRCG in a given channel. The double-Delta feature is calculated by applying equation (11) to the Delta feature.

The calculation of the  $U$ -dimensional cochleagram feature in Fig. 3A is detailed in Fig. 3C. We first filter input noisy speech by the 64-channel gammatone filterbank, then calculate the energy of each time-frequency unit by  $\sum_{k=1}^K s_{c,k}^2$  given the frame length  $K$ , and finally rescale the energy by  $\log_{10}(\cdot)$ , where  $s_{c,k}$  represents the  $k$ th sample of a given frame in the  $c$ th channel [40].

Note that when MRCG is used for bDNN training, it should be normalized to zero means and unit standard deviations in dimension *globally*, and the normalization factors should be used to normalize each test frame, where the word “globally” means that the normalization is conducted on the entire training corpus but not on each training utterance separately.

#### V. EVALUATION RESULTS OF NOISE-DEPENDENT MODELS

The term *noise-dependent* (ND) means that the noise scenarios of the training and test sets of machine-learning-based models are the same in terms of noise type and SNR level.

In this section, we first report the results of the proposed methods in Section V-B, and then analyze how different settings of MRS, bDNN, and MRCG improve the performance over their comparison methods in Section V-C.

##### A. Experimental Settings

1) *Data Sets*: We used the noisy speech corpora of ARURORA2 [28] as well as the clean speech corpus of AURORA4 [29] mixed with the NOISEX-92 noise corpus [37] for evaluation. AURORA2 contains the pronunciations of digits. AURORA4 contains the utterances of continuous speech. Note that the experimental conclusions in AURORA2 and AURORA4 are consistent. The data sets were preprocessed as follows.

We used 7 noisy test sets of AURORA2 [28] in SNR levels of  $[-5, 0, 5, 10, 15, 20]$  dB for an overall comparison between the proposed method and the competitive methods. That is to say, we had 42 noisy environments for the overall evaluation. The sampling rate is 8 kHz. The preprocessing of each test corpus was similar with that in [45]. Specifically, we split each test corpus to three subsets for training, developing, and test, each of which contains 300, 300, and 401 utterances respectively. All short utterances in each set were concatenated to a long conversation for simulating the real working environment of VAD.

We used the clean speech corpus of AURORA4 [29] corrupted by the ‘babble’ and ‘factory’ noise in the NOISEX-92 noise corpus in extremely low SNR levels (i.e.  $[-5, 0, 5]$

<sup>2</sup>Code is downloadable from <http://web.cse.ohio-state.edu/pnl/software.html>



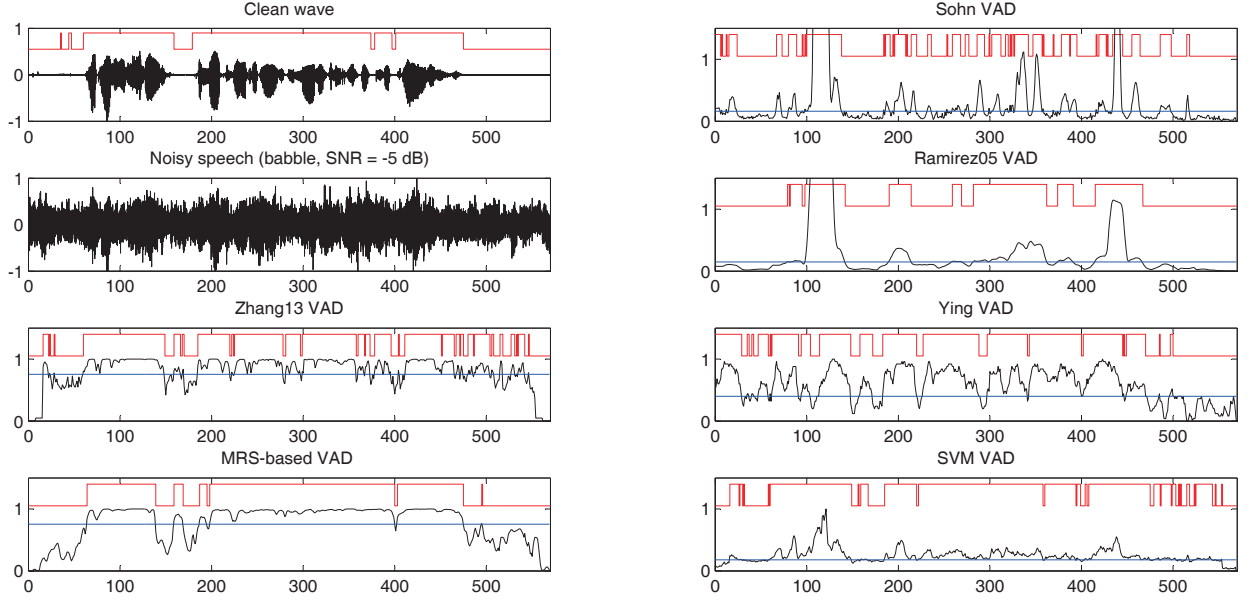


Fig. 4. Illustration of the proposed and comparison methods in the babble noise environment with  $\text{SNR} = -5$  dB. The soft outputs of all methods have been normalized so as to be shown clearly in the range  $[0, 1]$ . The straight lines are the optimal decision thresholds (tuned on the entire development corpus) in terms of HIT-FA, and the notched lines show the hard decisions on the soft outputs.

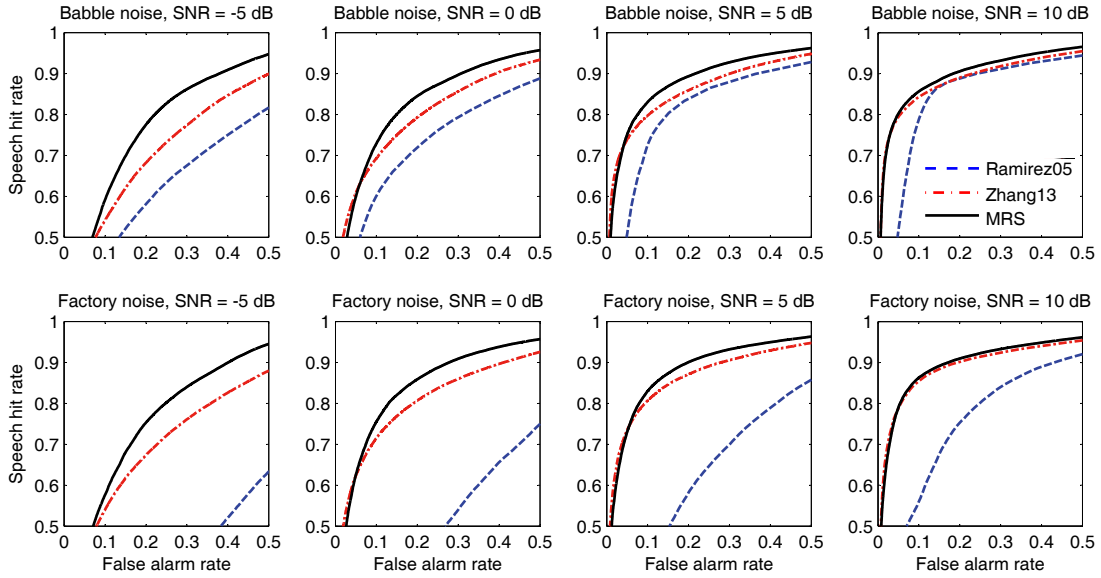


Fig. 5. ROC curve comparison between the proposed method and some representative VADs in 6 noise environments of the AURORA4 corpus.

dB) for a more broaden and harsh comparison between the proposed method and the competitors and for an investigation of the effectiveness of the components of the proposed method. That is to say, we constructed 6 difficult noisy speech corpora. The sampling rate is 16 kHz. The preprocessing is as follows. The clean speech corpus consists of 7,138 training utterances and 330 test utterances. We randomly selected 300 and 30 utterances from the training utterances as our training set and development set respectively, and used all 330 test utterances for testing. Note that for each noisy corpora, the additive noises for training, development, and test were cut from different intervals of a given noise.

The ground-truth labels of each noisy speech corpus in

either AURORA2 or AURORA4 were the results of Sohn's VAD [34] applied to the corresponding clean speech corpus. The frame length and frame shift of the proposed method were described in the MRCG feature. The frame length and frame shift of all competitive methods were 25ms and 10ms respectively. Note that the training set in AURORA4 is much larger than that in AURORA2.

2) *Evaluation Metrics*: ROC curve was used as the main metric. Its corresponding AUC was also reported. In addition, HIT-FA of the optimal operating point on the ROC curve was reported, where the optimal operating point is defined as a decision threshold achieving the highest HIT-FA (defined in Eq. (10)) on the development set. Because over 70% frames

TABLE I

AUC (%) COMPARISON BETWEEN THE COMPARISON VADS AND PROPOSED bDNN- AND MRS-BASED VADS ON THE AURORA2 CORPUS. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS. “REST.” IS SHORT FOR RESTAURANT.

Noise	SNR	Sohn	Ramirez05	Ying	SVM	Zhang13	bDNN	MRS
Babble	−5 dB	60.45	61.33	59.17	70.14	72.21	81.55	<b>82.51</b>
	0 dB	68.66	70.08	65.63	79.91	83.28	89.03	<b>89.85</b>
	5 dB	79.83	81.94	76.52	88.14	89.99	92.72	<b>92.93</b>
	10 dB	86.76	88.12	84.46	91.86	94.07	94.18	<b>94.84</b>
	15 dB	90.31	90.60	88.71	93.58	95.33	95.21	<b>95.36</b>
	20 dB	93.08	91.57	91.97	94.65	95.73	95.76	<b>95.85</b>
Car	−5 dB	59.03	60.62	61.75	82.18	82.76	91.34	<b>92.40</b>
	0 dB	69.05	72.00	69.27	89.50	91.64	94.87	<b>95.56</b>
	5 dB	79.83	82.22	78.53	93.59	93.96	95.60	<b>96.30</b>
	10 dB	87.22	88.64	84.73	95.01	95.69	96.30	<b>96.98</b>
	15 dB	91.53	92.16	89.41	96.05	96.45	96.97	<b>97.40</b>
	20 dB	94.09	92.82	92.87	96.78	97.33	97.16	<b>97.67</b>
Rest.	−5 dB	55.62	55.04	57.67	72.01	74.20	82.40	<b>84.03</b>
	0 dB	66.00	64.52	62.98	81.11	81.14	88.07	<b>89.81</b>
	5 dB	72.24	73.65	71.54	89.25	91.01	93.13	<b>94.20</b>
	10 dB	79.85	80.85	79.29	91.78	93.25	94.80	<b>95.45</b>
	15 dB	86.10	87.36	84.70	93.43	94.61	95.60	<b>96.29</b>
	20 dB	89.27	89.23	88.99	94.92	95.57	96.13	<b>96.73</b>
Street	−5 dB	53.68	54.80	55.63	72.77	74.32	85.57	<b>86.31</b>
	0 dB	60.03	60.06	61.62	81.29	81.38	89.22	<b>90.23</b>
	5 dB	68.74	71.52	70.28	88.03	90.01	92.53	<b>93.06</b>
	10 dB	76.04	78.21	76.41	91.19	92.39	93.95	<b>94.21</b>
	15 dB	83.35	86.59	82.27	92.61	94.69	94.23	<b>94.66</b>
	20 dB	88.73	88.96	87.14	94.15	95.31	94.89	<b>95.14</b>
Airport	−5 dB	56.60	59.39	59.06	73.59	76.30	82.77	<b>85.02</b>
	0 dB	64.22	66.11	66.07	81.97	83.38	90.87	<b>92.00</b>
	5 dB	73.78	76.90	74.48	88.90	89.59	94.29	<b>95.22</b>
	10 dB	83.18	86.06	84.21	92.61	94.35	95.82	<b>96.39</b>
	15 dB	89.62	90.31	88.43	95.14	96.11	96.52	<b>96.96</b>
	20 dB	91.27	90.05	91.58	95.68	96.88	97.15	<b>97.53</b>
Train	−5 dB	55.31	57.68	61.35	74.43	76.55	85.55	<b>86.82</b>
	0 dB	60.04	63.19	67.85	83.21	84.89	90.49	<b>91.68</b>
	5 dB	73.00	77.26	77.58	89.72	91.12	93.54	<b>94.60</b>
	10 dB	83.76	84.51	82.18	92.07	93.29	94.43	<b>95.21</b>
	15 dB	87.50	89.17	87.30	94.10	95.13	95.28	<b>96.18</b>
	20 dB	91.78	90.61	91.00	94.84	95.34	95.64	<b>96.55</b>
Subway	−5 dB	55.42	55.00	57.74	83.36	84.29	92.37	<b>93.28</b>
	0 dB	62.66	61.63	62.75	89.22	90.11	93.90	<b>94.23</b>
	5 dB	70.49	76.50	68.35	91.86	93.09	95.18	<b>95.63</b>
	10 dB	79.02	81.18	76.73	93.32	93.97	95.34	<b>95.95</b>
	15 dB	84.43	87.40	82.72	94.22	94.84	95.68	<b>96.17</b>
	20 dB	87.23	88.93	85.99	94.81	95.38	95.92	<b>96.43</b>

are speech, we did not use detection accuracy as the evaluation metric, so as to prevent reporting misleading results caused by class imbalance.

3) *Comparison Methods and Parameter Settings:* We compared bDNN- and MRS-based VADs with 5 VADs—Sohn VAD [34], Ying VAD [43], Zhang13 VAD [45], and SVM-based VAD, where the first two are the classic statistical model based ones and the last two are the more recent supervised learning based ones. The parameters of the referenced methods were well-tuned according to the authors.

For bDNN-based VAD, the parameter setting of bDNN was as follows. The numbers of hidden units were set to 512 for both the first and second hidden layers. The number of epoches was set to 50. The batch size was set to 512, the scaling factor for the adaptive stochastic gradient descent was set to 0.0015, and the learning rate decreased linearly from 0.08 to 0.001. The momentum of the first 5 epoches was set to 0.5, and the momentum of other epoches was adjusted to 0.9. The dropout

TABLE II

HIT-FA (%) COMPARISON BETWEEN THE COMPARISON VADS AND PROPOSED bDNN- AND MRS-BASED VADS ON THE AURORA2 CORPUS. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS. “REST.” IS SHORT FOR RESTAURANT.

Noise	SNR	Sohn	Ramirez05	Ying	SVM	Zhang13	bDNN	MRS
Babble	−5 dB	15.61	17.14	13.91	30.60	33.18	45.14	<b>49.71</b>
	0 dB	27.18	30.24	23.38	46.63	51.59	60.96	<b>64.11</b>
	5 dB	45.88	50.29	40.79	61.63	64.52	70.20	<b>71.86</b>
	10 dB	59.93	63.23	55.34	68.67	72.81	73.31	<b>74.11</b>
	15 dB	67.35	69.51	63.28	72.60	76.37	76.36	<b>76.99</b>
	20 dB	70.56	71.90	70.46	74.47	77.49	76.42	<b>77.56</b>
Car	−5 dB	14.22	15.96	17.96	50.61	51.93	66.78	<b>69.39</b>
	0 dB	29.83	34.58	30.38	64.62	68.96	75.22	<b>77.17</b>
	5 dB	47.02	51.23	43.82	72.87	73.07	76.66	<b>78.90</b>
	10 dB	61.20	64.88	55.17	75.52	76.84	78.86	<b>80.65</b>
	15 dB	70.73	71.43	64.68	78.27	78.75	80.62	<b>82.23</b>
	20 dB	76.74	72.97	72.05	80.23	81.80	81.65	<b>83.29</b>
Rest.	−5 dB	8.46	8.39	10.89	32.49	35.91	46.13	<b>50.22</b>
	0 dB	23.81	22.19	18.23	48.74	48.20	60.72	<b>64.79</b>
	5 dB	32.94	36.01	32.87	64.11	66.81	72.42	<b>74.86</b>
	10 dB	47.39	49.82	44.92	69.59	72.87	76.72	<b>78.34</b>
	15 dB	59.25	62.85	54.25	73.81	76.69	78.28	<b>80.75</b>
	20 dB	66.09	67.84	62.66	77.40	78.69	80.55	<b>82.04</b>
Street	−5 dB	5.50	8.30	9.42	34.34	35.81	55.42	<b>57.35</b>
	0 dB	16.28	16.39	16.19	47.78	47.59	63.19	<b>65.86</b>
	5 dB	28.96	34.05	30.96	61.21	64.52	70.67	<b>72.05</b>
	10 dB	40.20	43.62	39.92	67.36	69.55	73.91	<b>75.20</b>
	15 dB	54.31	60.05	51.24	71.13	74.27	74.14	<b>75.80</b>
	20 dB	64.43	64.42	60.79	74.17	76.99	76.33	<b>76.94</b>
Airport	−5 dB	8.80	15.35	13.83	35.29	39.49	49.14	<b>53.01</b>
	0 dB	22.93	25.86	26.02	50.25	52.62	66.42	<b>68.59</b>
	5 dB	36.73	42.92	37.73	62.99	63.92	73.90	<b>75.89</b>
	10 dB	52.96	60.35	54.72	70.05	73.78	77.79	<b>79.32</b>
	15 dB	63.50	68.93	62.99	76.01	78.18	79.78	<b>80.92</b>
	20 dB	69.27	68.81	68.88	77.35	80.45	81.71	<b>82.95</b>
Train	−5 dB	9.75	10.89	15.98	37.11	39.96	54.93	<b>57.34</b>
	0 dB	16.50	19.31	27.07	52.51	55.05	65.20	<b>68.99</b>
	5 dB	36.20	45.08	41.79	65.29	67.10	72.45	<b>75.40</b>
	10 dB	54.10	58.47	50.87	70.55	72.75	74.60	<b>76.90</b>
	15 dB	61.00	64.87	60.29	75.41	76.66	76.57	<b>79.62</b>
	20 dB	70.69	69.87	68.19	75.90	76.67	77.96	<b>80.46</b>
Subway	−5 dB	8.23	9.69	11.19	52.82	54.16	69.35	<b>71.42</b>
	0 dB	19.69	18.99	16.85	64.11	66.75	74.26	<b>75.37</b>
	5 dB	30.66	41.90	25.76	71.37	74.11	77.63	<b>80.06</b>
	10 dB	45.53	49.98	39.32	75.27	76.81	78.95	<b>80.68</b>
	15 dB	55.70	61.81	49.59	75.92	77.96	79.34	<b>81.51</b>
	20 dB	61.84	66.19	56.57	77.21	79.40	80.32	<b>82.21</b>

TABLE III

AUC (%) COMPARISON BETWEEN THE COMPARISON VADS AND PROPOSED bDNN-BASED AND MRS-BASED VADS ON THE AURORA4 CORPUS. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS.

Noise	SNR	Sohn	Ramirez05	Ying	SVM	Zhang13	bDNN	MRS
Babble	−5 dB	70.69	75.90	64.63	81.05	82.84	85.75	<b>86.60</b>
	0 dB	77.67	83.05	70.72	86.06	88.33	89.62	<b>90.15</b>
	5 dB	84.53	87.85	78.70	90.49	91.61	92.75	<b>93.02</b>
	10 dB	89.18	89.93	85.61	91.05	93.01	93.81	<b>93.93</b>
Factory	−5 dB	58.17	58.37	62.56	78.63	81.81	85.78	<b>85.81</b>
	0 dB	64.56	67.21	68.79	86.05	88.39	90.64	<b>90.76</b>
	5 dB	72.92	76.82	75.83	89.10	91.72	92.82	<b>92.98</b>
	10 dB	80.80	84.72	82.64	92.21	93.13	93.64	<b>93.69</b>

rate of the hidden units was set to 0.2. The half-window size  $W$  was set to 19, and the parameter  $u$  of the window was set to 9.

For MRS-based VAD, we trained two stacks (i.e. parameter  $S = 2$ ). For the bottom stack, we trained 10 bDNNs with resolution parameter  $(W, u)$  set to  $[(3, 1), (5, 2), (9, 4), (13, 6),$

TABLE IV

HIT-FA (%) COMPARISON BETWEEN THE COMPARISON VADS AND PROPOSED bDNN-BASED VAD ON THE AURORA4 CORPUS. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS.

Noise	SNR	Sohn	Ramirez05	Ying	SVM	Zhang13	bDNN	MRS
Babble	-5 dB	29.44	38.45	21.03	45.69	48.33	56.13	57.92
	0 dB	40.64	52.09	29.76	56.31	60.01	63.94	65.15
	5 dB	54.42	65.23	42.70	67.77	69.94	72.04	73.10
	10 dB	67.50	70.89	56.12	69.75	74.75	75.74	76.03
Factory	-5 dB	12.00	13.43	19.50	42.11	47.42	54.60	55.51
	0 dB	21.04	25.63	28.42	56.93	62.00	66.64	67.18
	5 dB	33.40	40.11	38.83	64.19	70.72	72.25	73.18
	10 dB	47.33	55.39	50.47	73.36	75.66	75.86	76.44

TABLE V

AUC (%) ANALYSIS OF THE RELATIVE CONTRIBUTIONS OF bDNN, MRS, AND MRCG. “COMB” DENOTES A SERIAL COMBINATION OF 11 ACOUSTIC FEATURES IN [45].

Noise	SNR	DNN+ COMB	bDNN+ COMB	MRS+ COMB	DNN+ MRCG	bDNN+ MRCG	MRS+ MRCG
Babble	-5 dB	81.53	84.62	86.11	81.54	85.75	86.60
	0 dB	85.48	88.84	89.76	86.48	89.62	90.15
	5 dB	89.08	92.11	92.82	90.05	92.75	93.02
	10 dB	90.56	93.10	93.65	91.64	93.81	93.93
Factory	-5 dB	80.16	83.51	85.75	79.70	85.78	85.81
	0 dB	84.59	88.95	90.35	86.51	90.64	90.76
	5 dB	87.79	91.91	92.70	89.76	92.82	92.98
	10 dB	89.16	92.79	93.75	90.95	93.64	93.69

(15, 7), (17, 8), (19, 9), (21, 10), (23, 11), (25, 12)] respectively. The parameter setting of each bDNN was exactly the same as that of the aforementioned bDNN-based VAD. For the top stack, we trained 1 bDNN with  $(W, u)$  set to (19, 9). The parameter setting of the bDNN was as follows. The numbers of hidden units were set to 128 for both the first and second hidden layers. The number of epoches was set to 7.

## B. Results

Tables I and II list the AUC and HIT-FA results of all 7 VAD methods on 42 noisy environments of AURORA2. Tables III and IV list the AUC and HIT-FA results on 8 noisy environments of AURORA4. Figure 4 illustrates the soft outputs of the MRS-based VAD as well as all comparison methods for the babble noise at -5 dB SNR. Figure 5 shows the ROC curve comparison between the MRS-based VAD, Ramirez05 VAD, and Zhang13 VAD on AURORA4. From the tables and figures, we observe that (i) the proposed method significantly outperforms all 5 others, particularly when the background is very noisy; (ii) the experimental phenomena of the proposed method on different noisy scenarios of AURORA2 and AURORA4 are quite consistent, which means its superiority is not affected by whether the speech utterances were continuous or isolated. Additionally, we find that AUC and HIT-FA match quite well.

## C. Effectiveness Evaluation of the Components of the MRS Based VAD

In this subsection, we evaluate the effectiveness of the components of the bDNN- and MRS-based VADs on the 8 noisy environments of AURORA4.

TABLE VI

HIT-FA (%) ANALYSIS OF THE RELATIVE CONTRIBUTIONS OF bDNN, MRS, AND MRCG. “COMB” DENOTES A SERIAL COMBINATION OF 11 ACOUSTIC FEATURES IN [45].

Noise	SNR	DNN+ COMB	bDNN+ COMB	MRS+ COMB	DNN+ MRCG	bDNN+ MRCG	MRS+ MRCG
Babble	-5 dB	49.15	53.58	55.75	46.14	56.13	57.92
	0 dB	57.60	61.81	63.44	55.79	63.94	65.15
	5 dB	66.40	70.58	72.47	65.06	72.04	73.10
	10 dB	71.83	74.22	75.21	70.28	75.74	76.03
Factory	-5 dB	45.82	51.30	55.29	42.32	54.60	55.51
	0 dB	54.54	62.36	65.64	55.82	66.64	67.18
	5 dB	64.46	70.50	72.45	63.88	72.25	73.18
	10 dB	69.17	73.83	76.09	68.21	75.86	76.44

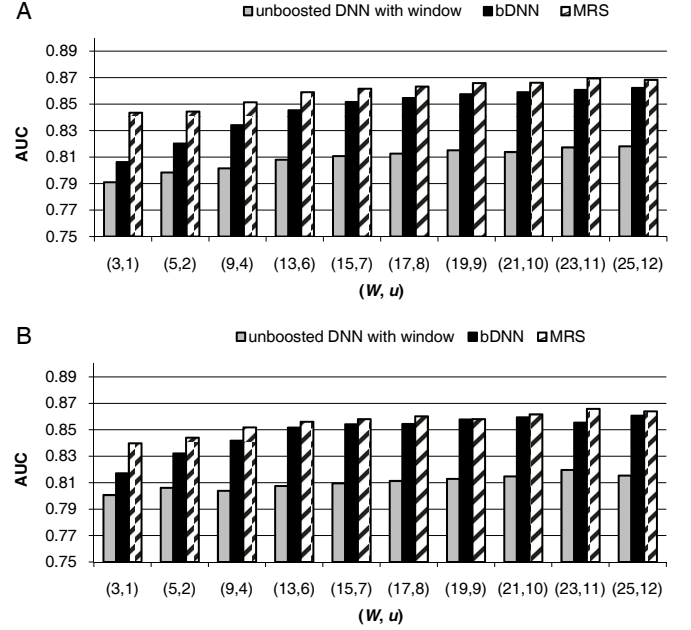


Fig. 6. AUC analysis of the advantage of the boosted algorithm in bDNN-based and MRS-based VADs over the unboosted counterpart that uses the same input  $\mathbf{x}'_n$  as bDNN and MRS but uses the original output  $y_n$  as the training target instead of  $\mathbf{y}'_n$ . (A) Comparison in the babble noise environment with SNR = -5 dB. (B) Comparison in the factory noise environment with SNR = -5 dB. Note that  $(W, u)$  are two parameters of the window of bDNN.

1) *Separate Effects of Boosted DNN, MRS, and MRCG on the Performance:* To separate the contributions of bDNN, MRS and MRCG to the performance improvement, we ran 6 experiments using either DNN, bDNN, or MRS as the model with either the combination (COMB) of 11 acoustic features in Zhang13 VAD [45] or MRCG as the input feature on the babble and factory noises in AURORA4, where the model “DNN” used the same DNN source code as that of bDNN with  $W$  set to 0. Tables V and VI list the AUC and HIT-FA comparisons between these 6 combinations. From the tables, we observe that (i) the 96-dimensional MRCG feature is at least as equivalently good as the 273-dimensional COMB feature; (ii) bDNN better than DNN; and (iii) MRS better than bDNN. Note that when we further enlarge the dimension of MRCG from 96 to 768 as we did in [47], MRCG can significantly outperform COMB (see [47] for the detailed results), though it is inefficient to do so in practice.

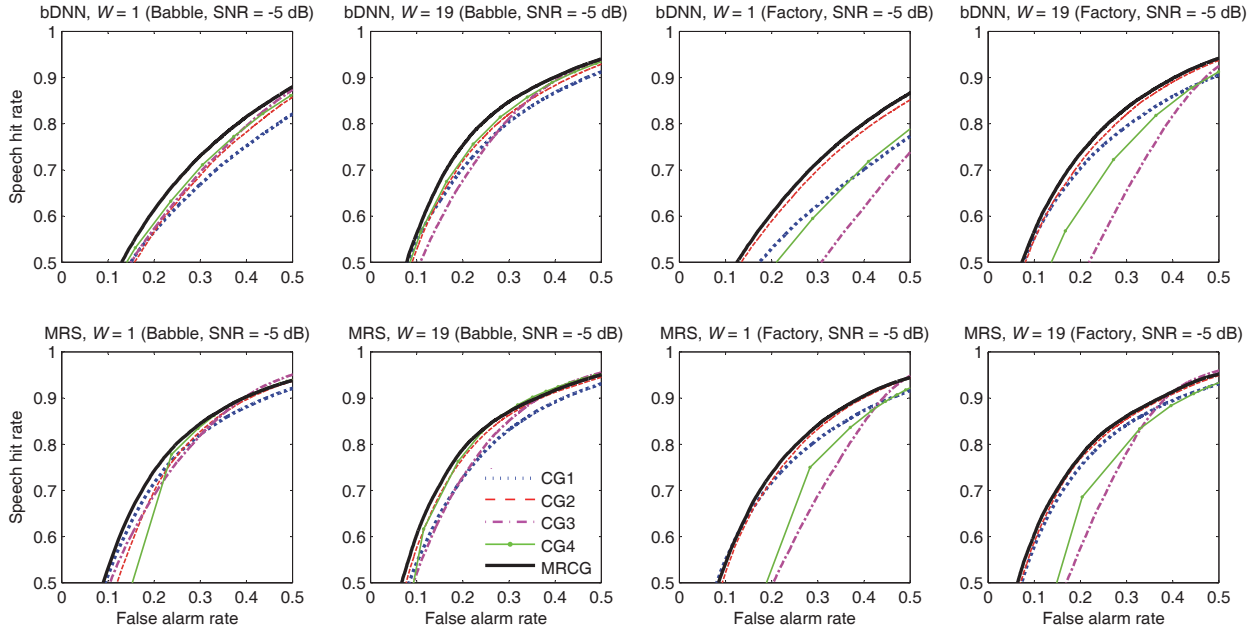


Fig. 8. ROC curve analysis of the MRCG feature versus its components at AURORA4.

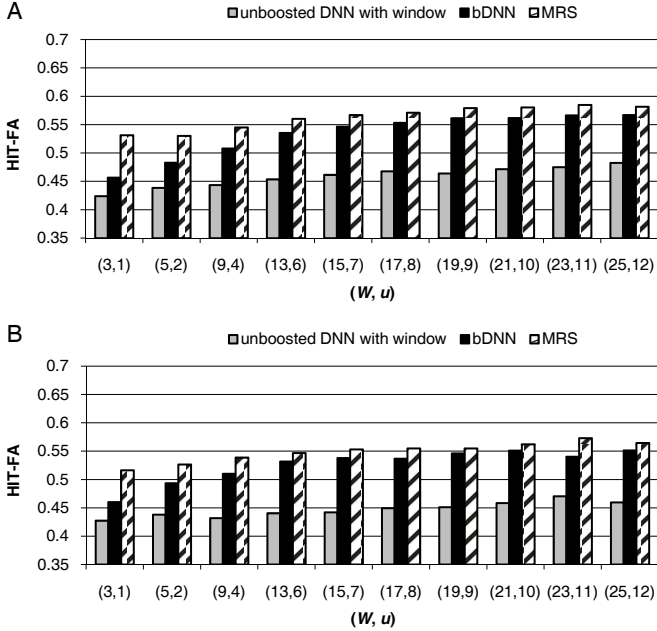


Fig. 7. HIT-FA analysis of the advantage of the boosted algorithm in bDNN-based and MRS-based VADs over the unboosted counterpart that uses the same input  $\mathbf{x}'_n$  as bDNN and MRS but uses the original output  $y_n$  as the training target instead of  $y'_n$ . (A) Comparison in the babble noise environment with SNR = -5 dB. (B) Comparison in the factory noise environment with SNR = -5 dB. Note that  $(W, u)$  are two parameters of the window of bDNN.

2) *Effects of Boosting*: To investigate how the boosted method is better than no boosting, we compared bDNN and MRS with a DNN model that used the same input as bDNN (i.e.,  $\mathbf{x}'_n$ ) but aimed to predict the label of only the central frame of the input (i.e.,  $y_n$ ) in terms of AUC (Fig. 6) and HIT-FA (Fig. 7) in the two difficult environments. Results show that (i) bDNN and MRS significantly outperforms the unboosted DNN, and their superiority becomes more and

more apparent when the window is gradually enlarged; (ii) the unboosted DNN can also benefit from the contextual information, but this performance gain is limited. Note that the boosted method had the same computational complexity with the unboosted one.

3) *Multi-resolution effects*: Figure 8 shows the ROC curve comparison between the MRCG feature and its four components in the two difficult noise environments with parameters  $(W, u)$  set to  $(0, 0)$  and  $(19, 9)$ , where  $W = 0$  means that bDNN reduces to DNN. From the figure, we observe that (i) MRCG is at least as good as the best of its 4 components in all cases, which demonstrates the effectiveness of the multi-resolution technique; (ii) CG2 yields a better ROC curve than the other 3 components. The same phenomena can also be observed when the dimension of MRCG is enlarged to 768 as shown in [47].

## VI. EVALUATION RESULTS OF NOISE-INDEPENDENT MODELS

The term *noise-independent* (NI) means that once trained, the machine learning based VADs can achieve reasonable performance in various noise scenarios, even though the noise scenarios are unseen from the training set. Training good NI models is one of the ultimate goals of machine learning based VADs in real-world applications and also one of the most difficult tasks that prohibit machine learning methods from practical use. In this section, we train such models and show their promising performance in difficult and unseen test scenarios.

### A. Experimental Settings

We randomly selected 300 clean utterances from AURORA2 and AURORA4 respectively as the clean corpora, which were also used as the clean corpora in Section V

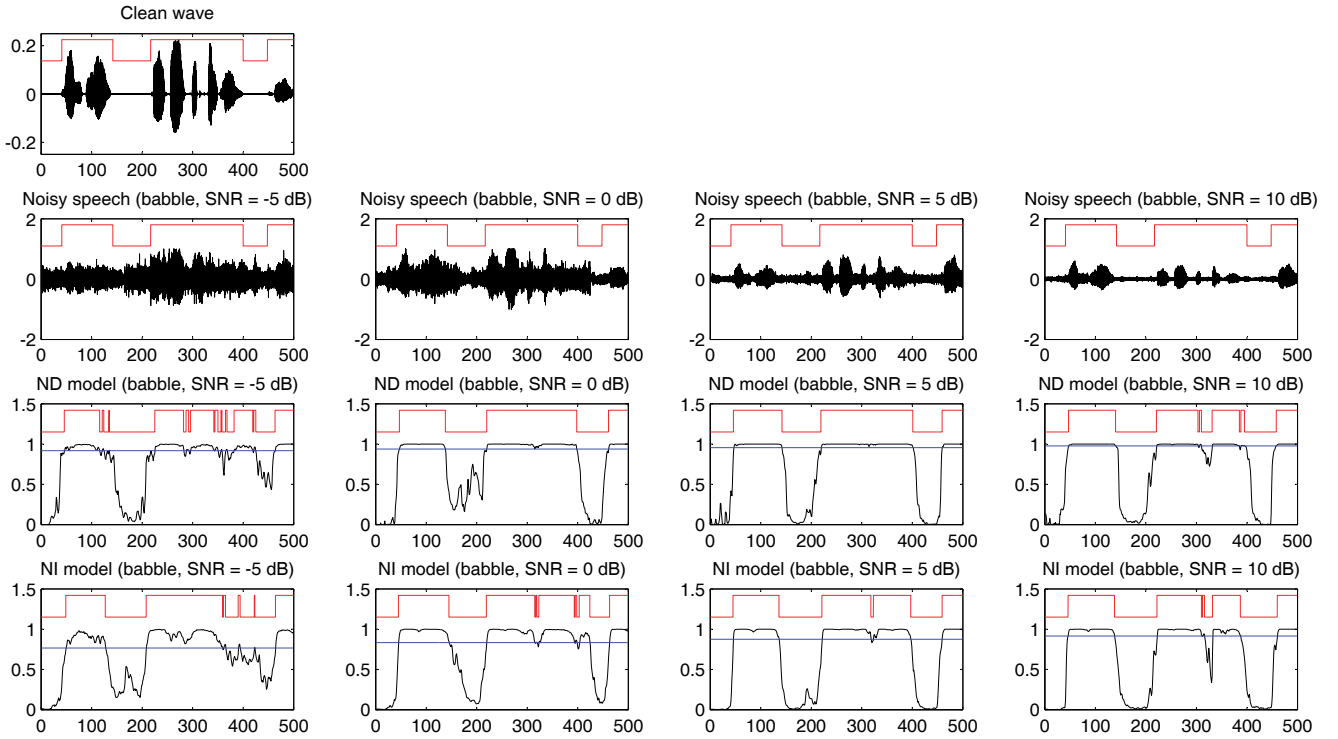


Fig. 9. Visualization of the output of noise-dependent (ND) MRS model and noise-independent (NI) MRS model in the babble noise environment at various SNR levels. Each test scenario of ND model is exactly the same as its training scenario. The test scenarios of NI model are unseen from its training corpus.

for synthesizing noisy speech corpora. We used a large-scale sound effect library<sup>3</sup> as our noise corpus, which contains over 20,000 sound effects. For constructing the noisy training corpus of AURORA2, we first randomly selected 4000 noise segments and concatenated them to a long noise wave which is about 35 hours long; then, we randomly picked clean utterances from the clean corpus of AURORA2 and added them one by one in time slot to the long noise wave with SNR levels varying in  $[-10, -9, -8, -7, -6, -4, -3, -2, -1, 1, 2, 3, 4, 6, 7, 8, 9, 11, 12]$  dB, where repeated selection of the clean utterances was allowed. Note that when synthesizing each noisy speech segment in the long noisy speech wave, we fixed the clean utterance and rescaled the noise segment. For constructing the noisy test corpora of AURORA2, we used the same test noisy corpora as in Section V, which contains 28 noisy scenarios with SNR levels ranging in  $[-5, 0, 5, 10]$  dB. We constructed the noisy training corpus of AURORA4 in the same way as that of AURORA2, and used the same noisy test corpora as in Section V for evaluation of the NI models. From the above description, it is clear that the test noise scenarios are unseen in the training corpora.

We trained 1 DNN-, 1 bDNN-, and 1 MRS-based VAD on the noisy training corpus of AURORA2, and evaluated the 3 NI models on all 28 test corpora. We conducted an experiment on AURORA4 in the same way as that on AURORA2. The parameter settings of the DNN, bDNN, and MRS models were the same as their corresponding ND models in Section V.

## B. Results

It was supposed that the ND models, which were trained and tested in the same noise scenarios, would perform better than NI models. In this section, we investigated how much their performance differed. Fig. 9 shows a visualized comparison of the soft output of NI model and 4 ND models on AURORA2. Tables VII and VIII list the AUC and HIT-FA comparison of NI and ND models of the DNN-, bDNN-, and MRS-based VADs on AURORA2. Tables IX and X list the comparison on AURORA4. From the figure and tables, we observe that (i) the performance of NI models approaches to and even outperforms the performance of ND models in most cases of AURORA2 when the SNR is equal or greater than 0 dB and in all cases of AURORA4; (ii) NI models perform slightly worse than ND models on AURORA2 when the SNR is extremely low, e.g.  $-5$  dB. Comparing the above four tables with Tables I, II, III, and IV in Section V, we observe that (i) MRS-based VAD with NI training performs significantly better than Zhang13 VAD and SVM VAD with ND training in most test cases of AURORA2 except the subway environment which is a very burst and different non-stationary noise [46]; (ii) the proposed VADs with NI training are also significantly better than Sohn VAD and Ramirez05 VAD whose main merit is that they can be used in various noise scenarios without offline training.

## VII. CONCLUDING REMARKS

In this paper, we have proposed a supervised VAD method, named MRS-based VAD, using a new base classifier—bDNN—and a newly introduced acoustic feature—MRCG. The proposed method explore contextual information heavily

<sup>3</sup>The library was requested from <http://www.sound-ideas.com/sound-effects/series-6000-combo-sound-effects.html>

TABLE VII

AUC (%) COMPARISON BETWEEN THE NOISE-INDEPENDENT (NI) MODELS AND NOISE-DEPENDENT (ND) MODELS AT AURORA2.

Noise	SNR	DNN		bDNN		MRS	
		NI	ND	NI	ND	NI	ND
Babble	−5 dB	73.36	78.62	76.95	81.55	77.92	82.51
	0 dB	86.28	85.82	89.36	89.03	90.16	89.85
	5 dB	92.65	89.07	94.27	92.72	94.32	92.93
	10 dB	94.49	88.85	95.88	94.18	95.76	94.84
Car	−5 dB	86.36	89.75	88.71	91.34	89.01	92.40
	0 dB	92.72	93.74	94.24	94.87	94.06	95.56
	5 dB	94.66	94.53	95.90	95.60	95.86	96.30
	10 dB	95.33	95.35	96.59	96.30	96.48	96.98
Restaurant	−5 dB	71.16	78.15	73.33	82.40	75.06	84.03
	0 dB	83.49	86.57	86.27	88.07	87.24	89.81
	5 dB	91.49	92.23	93.41	93.13	93.84	94.20
	10 dB	94.10	93.96	95.31	94.80	95.46	95.45
Street	−5 dB	79.50	82.89	80.88	85.57	81.34	86.31
	0 dB	89.78	87.77	91.36	89.22	91.13	90.23
	5 dB	93.39	89.90	94.56	92.53	94.14	93.06
	10 dB	94.62	91.18	95.84	93.95	95.60	94.21
Airport	−5 dB	77.34	80.54	79.78	82.77	81.04	85.02
	0 dB	88.09	89.58	90.22	90.87	90.38	92.00
	5 dB	93.08	92.97	94.63	94.29	94.58	95.22
	10 dB	94.82	94.59	96.24	95.82	96.16	96.39
Train	−5 dB	80.23	84.18	82.98	85.55	83.84	86.82
	0 dB	89.20	88.89	91.24	90.49	91.14	91.68
	5 dB	93.32	92.20	94.88	93.54	94.92	94.60
	10 dB	94.05	92.80	95.52	94.43	95.39	95.21
Subway	−5 dB	72.64	91.35	74.47	92.37	75.80	93.28
	0 dB	85.10	93.43	86.78	93.90	87.18	94.23
	5 dB	91.57	94.39	92.76	95.18	93.13	95.63
	10 dB	93.61	95.07	94.61	95.34	94.93	95.95

in three levels. At the top level, MRS is a stack of ensemble classifiers. The classifiers in a building block explore context in different resolutions and output different predictions which are further integrated in their upper building block. At the middle level, bDNN is a strong DNN classifier that first produces multiple base predictions on a single frame by boosting the contextual information encoded in a given resolution and then aggregates the base predictions for a stronger one. At the bottom level, MRCG consists of cochleagram features at multiple spectrotemporal resolutions. Experimental results on AURORA2 and AURORA4 have shown that when the noise scenarios of training and test are matching, the proposed method outperforms the referenced VADs by a considerable margin, particularly at low SNRs. Our further analysis shows that (i) both bDNN and MRS contribute to the improvement; (ii) the 96-dimensional MRCG feature is comparable to the 273-dimensional COMB feature. Moreover, when trained with a large number of noise scenarios and a wide range of SNR levels, the proposed method performs as good as the method with noise-dependent training, which is a promising sign for the practical use of machine-learning-based VADs in real-world environments.

#### ACKNOWLEDGMENTS

We thank Yuxuan Wang for providing his DNN version, Jitong Chen for providing the MRCG code and the large-scale sound effect library, and Arun Narayanan for helping with the AURORA4 corpus. Part of the work was conducted when

TABLE VIII

HIT−FA (%) COMPARISON BETWEEN THE NOISE-INDEPENDENT (NI) MODELS AND NOISE-DEPENDENT (ND) MODELS AT AURORA2.

Noise	SNR	DNN		bDNN		MRS	
		NI	ND	NI	ND	NI	ND
Babble	−5 dB	33.71	41.48	39.48	45.14	41.25	49.71
	0 dB	56.58	55.22	62.71	60.96	64.70	64.11
	5 dB	70.40	68.97	73.69	70.20	74.03	71.86
	10 dB	74.77	72.06	77.44	73.31	77.21	74.11
Car	−5 dB	56.98	62.49	61.74	66.78	63.02	69.39
	0 dB	70.63	71.68	74.06	75.22	74.08	77.17
	5 dB	75.67	74.48	78.66	76.66	78.82	78.90
	10 dB	77.40	76.71	80.01	78.86	79.65	80.65
Restaurant	−5 dB	31.43	41.41	34.76	46.13	37.91	50.22
	0 dB	51.48	54.33	57.51	60.72	59.90	64.79
	5 dB	68.21	67.93	73.30	72.42	74.35	74.86
	10 dB	75.16	73.45	78.18	76.72	78.49	78.34
Street	−5 dB	44.13	51.05	47.41	55.42	48.99	57.35
	0 dB	62.95	60.96	66.21	63.19	66.34	65.86
	5 dB	71.30	68.87	73.95	70.67	73.49	72.05
	10 dB	74.86	72.83	77.85	73.91	76.81	75.20
Airport	−5 dB	41.37	44.61	45.51	49.14	48.92	53.01
	0 dB	60.63	61.69	64.61	66.42	64.75	68.59
	5 dB	71.32	70.00	74.46	73.90	74.83	75.89
	10 dB	76.08	75.00	79.29	77.79	79.10	79.32
Train	−5 dB	45.77	50.31	51.00	54.93	53.26	57.34
	0 dB	62.81	60.95	66.54	65.20	67.59	68.99
	5 dB	72.77	68.84	76.77	72.45	76.73	75.40
	10 dB	74.58	72.07	78.08	74.60	77.63	76.90
Subway	−5 dB	33.29	65.09	38.14	69.35	40.92	71.42
	0 dB	55.85	71.52	60.47	74.26	60.94	75.37
	5 dB	69.61	75.09	72.91	77.63	73.40	80.06
	10 dB	76.03	75.93	78.64	78.95	79.25	80.68

TABLE IX

AUC (%) COMPARISON BETWEEN THE NOISE-INDEPENDENT (NI) MODELS AND NOISE-DEPENDENT (ND) MODELS AT AURORA4.

Noise	SNR	DNN		bDNN		MRS	
		NI	ND	NI	ND	NI	ND
Babble	−5 dB	78.79	81.54	81.65	85.75	84.09	86.60
	0 dB	84.31	86.48	86.55	89.62	88.32	90.15
	5 dB	88.90	90.05	90.40	92.75	91.19	93.02
	10 dB	91.28	91.64	92.83	93.81	93.19	93.93
Factory	−5 dB	78.58	79.70	81.20	85.78	83.40	85.81
	0 dB	84.52	86.51	86.81	90.64	88.34	90.76
	5 dB	89.25	89.76	90.99	92.82	91.46	92.98
	10 dB	91.21	90.95	92.94	93.64	93.16	93.69

the authors were visiting the Center of Intelligent Acoustics and Immersive Communications, Northwestern Polytechnical University, China. We also thank the Ohio Supercomputing Center for providing computing resources. The research was supported in part by an AFOSR grant (FA9550-12-1-0130).

#### REFERENCES

- [1] G. Anceja and B. Yegnanarayana, “Single frequency filtering approach for discriminating speech and nonspeech,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 4, pp. 705–717, 2015.
- [2] A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, “ITU-T Recommendation G. 729 Annex B: a silence compression scheme for use with G. 729 optimized for V. 70 digital simultaneous voice and data applications,” *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 64–73, 1997.
- [3] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 2, pp. 113–120, 1979.



TABLE X

HIT-FA (%) COMPARISON BETWEEN THE NOISE-INDEPENDENT (NI) MODELS AND NOISE-DEPENDENT (ND) MODELS AT AURORA4.

Noise	SNR	DNN		bDNN		MRS	
		NI	ND	NI	ND	NI	ND
Babble	-5 dB	42.44	46.14	47.14	56.13	51.51	57.92
	0 dB	51.04	55.79	55.36	63.94	58.81	65.15
	5 dB	62.10	65.06	64.75	72.04	66.24	73.10
	10 dB	69.26	70.28	71.90	75.74	72.48	76.03
Factory	-5 dB	42.41	42.32	48.13	54.60	51.77	55.51
	0 dB	51.83	55.82	56.61	66.64	59.88	67.18
	5 dB	63.23	63.88	66.68	72.25	67.11	73.18
	10 dB	69.81	68.21	73.32	75.86	73.40	76.44

- [4] L. Breiman, "Bagging predictors," *Machine Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [5] —, "Random forests," *Machine Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [6] J. H. Chang, N. S. Kim, and S. K. Mitra, "Voice activity detection based on multiple statistical models," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 1965–1976, 2006.
- [7] J. Chen, Y. Wang, and D. L. Wang, "A feature study for classification-based speech separation at very low signal-to-noise ratio," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 7089–7093.
- [8] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8609–8613.
- [9] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, 2012.
- [10] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, P. A. Tucker *et al.*, "Large scale distributed deep networks," in *Adv. Neural Inform. Process. Sys.*, 2012, pp. 1232–1240.
- [11] D. Enqing, L. Guizhong, Z. Yatong, and Z. Xiaodi, "Applying support vector machines to voice activity detection," in *Proc. Int. Conf. Signal Process.*, vol. 2, 2002, pp. 1124–1127.
- [12] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 32, no. 6, pp. 1109–1121, 1984.
- [13] ETSI, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," *ETSI ES*, vol. 202, no. 050.
- [14] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 483–487.
- [15] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. Int. Conf. Comput. Learn. Theory*, 1995, pp. 23–37.
- [16] S. Gazor and W. Zhang, "A soft voice activity detector based on a Laplacian-Gaussian model," *IEEE Trans. Speech, Audio Process.*, vol. 11, no. 5, pp. 498–505, 2003.
- [17] J. Gorritz, J. Ramirez, E. Lang, and C. Puntonet, "Hard c-means clustering for voice activity detection," *Speech Commun.*, vol. 48, no. 12, pp. 1638–1649, 2005.
- [18] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [19] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, pp. 1–18, 2012.
- [20] G. Hu and D. L. Wang, "Auditory segmentation based on onset and offset analysis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 2, pp. 396–405, 2007.
- [21] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7378–7382.
- [22] I. V. McLoughlin, "The use of low-frequency ultrasound for voice activity detection," in *Proc. Interspeech*, 2014, pp. 1553–1557.
- [23] S. Mousazadeh and I. Cohen, "Voice activity detection in presence of transient noise using spectral clustering," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 6, pp. 1261–1271, 2013.
- [24] S. K. Nemala, K. Patil, and M. Elhilali, "A multistream feature framework based on bandpass modulation filtering for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 416–426, 2013.
- [25] E. Nemer, R. Goubran, and S. Mahmoud, "Robust voice activity detection using higher-order statistics in the LPC residual domain," *IEEE Trans., Speech, Audio Process.*, vol. 9, no. 3, pp. 217–231, 2001.
- [26] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, 2012, pp. 1969–1972.
- [27] S. Nie, H. Zhang, X. Zhang, and W. Liu, "Deep stacking networks with time series for speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6667–6671.
- [28] D. Pearce, H. Hirsch *et al.*, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ICSLP'00*, vol. 4, 2000, pp. 29–32.
- [29] D. Pearce and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU384/02," Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep., 2002.
- [30] J. Ramirez, J. C. Segura, C. Benítez, L. García, and A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *IEEE Signal Process. Lett.*, vol. 12, no. 10, pp. 689–692, 2005.
- [31] S. O. Sadjadi and J. H. L. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 197–200, 2013.
- [32] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, 2013, pp. 3497–3501.
- [33] J. W. Shin, J. H. Chang, and N. S. Kim, "Voice activity detection based on statistical models and machine learning approaches," *Computer Speech & Lang.*, vol. 24, no. 3, pp. 515–530, 2010.
- [34] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, 1999.
- [35] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Machine Learn.*, 2013, pp. 1–8.
- [36] P. Teng and Y. Jia, "Voice activity detection via noise reducing using non-negative sparse coding," *IEEE Signal Process. Lett.*, vol. 20, no. 5, pp. 475–478, 2013.
- [37] The Rice University, "Noisex-92 database," <http://spib.rice.edu/spib>.
- [38] S. Thomas, G. Saon, M. Van Segbroeck, and S. S. Narayanan, "Improvements to the IBM speech activity detection system for the DARPA RATS program," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4500–4504.
- [39] K. Walker and S. Strassel, "The RATS radio traffic collection system," in *Proc. ISCA Odyssey*, 2012, pp. 291–297.
- [40] D. L. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. Wiley-IEEE Press, 2006.
- [41] Y. Wang, J. Chen, and D. L. Wang, "Deep neural network based supervised speech segregation generalizes to novel noises through large-scale training," Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, USA, Tech. Rep. OSU-CISRC-3/15-TR02, 2015.
- [42] Y. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, 2013.
- [43] D. Ying, Y. Yan, J. Dang, and F. Soong, "Voice activity detection based on an unsupervised learning framework," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2624–2644, 2011.
- [44] D. Yu, L. Deng, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 388–396, 2013.
- [45] X.-L. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 697–710, 2013.
- [46] X.-L. Zhang, "Unsupervised domain adaptation for deep neural network based voice activity detection," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2014, pp. 6864–6868.
- [47] X.-L. Zhang and D. L. Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection," in *Proc. Interspeech*, 2014, pp. 1534–1538.